

Jukka Vuorinen

WWW.KAIKILLE.FI – SIVUSTON LUOMINEN

Tietotekniikan koulutusohjelma
2014

WWW.KAIKILLE.FI – SIVUSTON LUOMINEN

Vuorinen, Jukka
Satakunnan ammattikorkeakoulu
Tietotekniikan koulutusohjelma
Huhtikuu 2014
Ohjaaja: Trast, Ismo
Sivumäärä: 31
Liitteitä: 8

Asiasanat: WordPress, esteetön, responsiivinen, dynaaminen, Ajax

Tämän opinnäytetyön tarkoituksena oli esteettömyyttä koskevan tiedon ja materiaalin levittämiseen tarkoitettun websivuston luominen. Tavoitteena oli esteetön ja nykyaikainen toteutus, joka käyttäisi WordPressiä sisällönhallintaan siten laajentaen esteettömyyden myös sisällöntuottajiin.

Tavoite saavutettiin tutkimalla WordPressin ohjelmointirajapinnan toimintaa. Sivusto rakennettiin käyttämällä nykyisiä HTML 5 ja CSS 3 standardeja, sekä php ja JavaScript kieliä.

Sivustoa testattiin Internet Explorer versioilla 9-11, simuloimalla toimintaa eri selainversioissa Internet Exploreriin rakennetun DOM Explorerin avulla sekä usealla eri Chromen ja Firefoxin versiolla ja Safarilla. Toimintaa testattiin myös mobiililaitteilla. Koska sivusto käyttää uudempia standardeja, se ei siksi tule toimimaan vanhemmilla selaimilla. käyttäjäkokemukset ovat olleet pääsääntöisesti positiivisia.

THE CREATION OF WWW.KAIKILLE.FI

Vuorinen, Jukka

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in Information Technology

April 2014

Supervisor: Trast, Ismo

Number of pages: 31

Appendices: 8

Keywords: WordPress, accessibility, responsive, dynamic, Ajax

The purpose of this thesis was to create a website for the purpose of spreading information and material about accessibility. The goal was accessible and modern website that would use the WordPress in content management thus extending accessibility to the content producers.

Goal was achieved by researching the WordPress API. Website itself was built by using the current HTML 5 and CSS 3 standards along with php and JavaScript languages.

Website was tested with the Internet Explorer versions 9-11, simulating the function of different browser versions with the DOM Explorer that is inbuilt to the newest Internet Explorer versions. Website was also tested with different versions of Chrome and Firefox and also with Safari. The function was also tested with different mobile devices. Because the site uses newest standards it does not therefore operate in older browser versions at all. The user tests have been mainly favorable.

SISÄLLYS

LYHENTEET	5
1 JOHDANTO.....	6
2 VERKKOSIVUJEN ESTEETTÖMYYS.....	6
2.1 Esteettömyyden määritelmä.....	6
2.2 Esteettömyyden arviointi	7
2.3 Toteutustekniikat.....	9
3 TOIMINNALLISET VAATIMUKSET.....	10
4 KEHITYSYMPÄRISTÖ.....	10
5 VERKKOSIVUJEN RESPONSIIVISUUS	11
5.1 Määritelmä	11
5.2 Bootstrap	12
5.3 Viewport meta.....	12
5.4 Rivinpituuden responsiivisuudesta	13
5.5 Kuvankoon responsiivisuus	13
6 VERKKOSIVUJEN DYNAAMISUUS.....	14
6.1 Määritelmä	14
6.2 Käytännön toiminta	14
6.3 Index.html	16
6.4 Disclaimer.html.....	17
6.5 Script.js	18
6.6 Script_dont_mod.js	20
6.7 Backend_wp.php.....	22
6.8 Backend.php.....	23
6.9 Style.css	24
6.10 Menu.css	26
7 TESTAUS	26
7.1 Verkkosivujen esteettömyyden ja käytettävyyden arviointi	26
7.2 Kaikille.fi-sivuston käytettävyyden kehittäminen ja heuristinen arviointi ..	27
8 PARANNUSEHDOTUKSET	28
9 YHTEENVETO	29
LÄHTEET.....	30
LIITTEET	

LYHENTEET

API	Application Programming Interface
CSS	Cascading Style Sheet
DOM	Document Object Model
HTML	HyperText Markup Language
IE	Internet Explorer
JSON	JavaScript Object Notation
PHP	Hypertext Preprocessor
RWD	Responsive Web Design
WAI	Web Accessibility Initiative
W3C	World Wide Web Consortium

1 JOHDANTO

Esteettömyys on yhä suuremmassa roolissa kotona, kouluissa, työpaikoilla sekä vapaa-aikana. Esteetön ympäristö on toimiva kaikille iästä ja toimintakyvystä riippumatta. Esteettömyys tarkoittaa myös palveluiden helppoa saavutettavuutta. Internet on nykyään kiinteä osa ympäristöämme, siksi myös palveluidemme tulisi olla verkossakin esteettömiä.

Internetsivujen heikko käytettävyys ja epäloogisuus johtuvat monesti heikosta sisältöarkkitehtuurista. Aivan kuten rakennetussakin ympäristössä, myös virtuaalisessa ympäristössä arkkitehtuuri muodostuu erilaisten tilojen ja alueiden (sivut), rakenteiden (informaatio) ja kulkuväylien (linkit) yhdistelmistä (Kuusinen & Tupala 2013, 87).

Projektissamme www.kaikille.fi, suunnittelimme ja loimme interaktiivisen verkko-opimisympäristön, jonka tarkoituksena on luoda pedagoginen malli esteettömälle verkkopalvelulle (Kuusinen & Vuorinen 2014).

2 VERKKOSIVUJEN ESTEETTÖMYYS

2.1 Esteettömyyden määritelmä

Web esteettömyys tarkoittaa sitä, että ihmiset, joilla on erilaisia vammoja, voivat käyttää Webbiä. Tarkemmin, Web esteettömyys tarkoittaa sitä että ihmiset joilla on vammoja voivat hahmottaa, ymmärtää, navigoida ja olla vuorovaikutuksessa Webbin kanssa ja sitä että he voivat antaa oman panoksensa Webbiin (W3C WAI: Introduction to Web Accessibility 2005).

Web esteettömyys hyödyttää myös muita käyttäjiä mukaan lukien vanhemmat ihmiset, joilla kyvyt muuttuvat iästä johtuen (W3C WAI: Introduction to Web Accessibility 2005).

2.2 Esteettömyyden arviointi

Esteettömyyden arviointiin on monia erilaisia testejä. Arviointiin käytettäviä testejä ovat:

- Silmäilytestit.
- Tarkempi tarkastelu.
- Näppäilytestit.
- Tarkistaminen eri selainasetuksilla.
- Tarkistaminen eri selaimilla.
- Automaattiset tarkistukset.

Silmäilytestit voidaan toteuttaa selaimella kuin selaimella. Sivunpituuden tulisi olla kohtuullinen ja mahtua kuvaruudulle kerralla. Sisältö sivu voi olla pidempi, mutta sivu saattaa parantua jakamalla se erillisiksi sivuiksi. Selainsivun yläpalkissa näkyvän ulkoisen otsikon tulisi olla kuvaava ja kertoa mitä sivu käsittelee, vaikka itse sivua ei vilkaistaisi.

Sisältösivu on parhaimmillaan sellainen, että lukemalla otsikot saa hyvän käsityksen siitä, mitä sivu käsittelee. Jos suunnistus sivulla on paljon linkkejä, kannattaa huomioida, että ne on ryhmitelty ja ryhmät ovat otsikoitu.

Sivun sisällön tulisi olla myös nopeasti hahmotettavissa ja arvioitaessa tulee myös tarkistaa onko sivulla erikielisiä tekstejä, sillä sivuston ulkoasun tulee olla yhtenäinen. Hyvä on myös miettiä, että onko sivustolla mahdollisesti olevan animaation tai muun liikkuvan elementin mukana olo perusteltua.

Tarkemmassa tarkastelussa selvitetään miten hyvin sivu toimii ääneen luettuna. Samalla katsotaan sisällön kappalejako ja kieliasun luettavuus. Myös linkkien tulisi erottua muusta tekstistä tarpeeksi selvästi ja perättäisten linkkien tulisi erottua myös toisistaan selvästi.

Tarkistaa tulisi myös aineisto johon linkit viittaavat, vaatiko se erityisen ohjelman katseluun? Periaatteessa tämä selviää linkkiä seuraamalla, mutta linkkien yhteydessä

voisi olla huomautus esim. ”PDF-muotoinen”. Aineistolle voisi myös tarjota vaihtoehtoisia linkkiä, josta se löytyy esteettömämmässä muodossa. Linkkien tulisi myös aueta alkuperäiseen selainikkunaan.

Oikein tehty sivusto toimii ilman hiirtäkin ja parhaiten sen testaa pelkällä näppäimistöllä. Huomiota tulisi kiinnittää voiko sivun sisällä kulkea linkistä toiseen tab-näppäimen avulla ja onko järjestys looginen. Tarkista samalla mahdolliset tarpeettomat linkit.

Vakionavigointia eli sivun alun linkkikokoelmaa arvioitaessa, tulee katsoa voiko sen ohittaa näppäimistöllä. Myös sivulla olevien lomakkeiden täytön tulisi sujua pelkällä näppäimistöllä.

Tarkistettaessa sivustoa eri selain asetuksilla tulisi arvioida kirjasin koon sopivuus, onko se helposti luettavissa, muttei turhan suurta? Sivuston kirjasinkokoa tulee myös voida muuttaa, jos se ei testissä muutu voidaan selain pakottaa noudattamaan käyttäjän asettamaa kokoa.

Testaa sivu selaimen oletusväreillä, joka yleensä on musta ja valkoinen. Sääda ikkunan leveyttä tarkistaaksesi sivun skaalautuvuus. Sivun toimivuus tulisi myös testata ilman CSS määritteitä ja JavaScriptejä.

Sivusto tulisi myös testata erilaisilla selaimilla. Ota kuvat pois päältä ja katso näkyvätkö vaihtoehto tekstit. Onko sivu tällöin ymmärrettävä? Taulukot tulee myös kytkeä testattaessa pois päältä, jotta saadaan tilanne vastaamaan monien selainten ainoaa toiminta mahdollisuutta.

Testaa sivusto ilman evästeitä. Evästeet oikein käytettynä avustavat esteettömyyttä, mutta väärin käytettynä sivut eivät toimi lainkaan ilman. Kytke kehykset myös pois päältä, huonosti tehdyt kehykset ovat tyhjiä tai niillä näkyy teksti selaintuen riittämättömyydestä. Muodosta sivun linkeistä lista control + j komennolla ja tarkista onko linkkejä kohtuullinen (enintään muutamia kymmeniä) määrä.

Sivuston testaamiseen on myös automaattisia välineitä, joita ovat:

- HTML-syntaksin tarkistin, joka tarkistaa HTML koodin oikeaoppisuuden. Tämä ei kuitenkaan takaa sivuston toimivuutta.
- CSS Validaattori, joka tarkistaa tyylitiedostojen standardien noudattamisen, esim. W3C CSS Validator.
- Linkkien teknisen toimivuuden tarkistin, esim. Xenu's Link Sleuth.
- Wave, joka esittää sivun visuaalisessa muodossa (selaimessa) siten, että siitä ilmenee eräitä loogisen rakenteen perusasioita.
- Eräiden esteettömyyteen liittyvien piirteiden tarkistin, kuten A-Prompt. Huomioi huomautukset esteettömyyden tarkistamisen välineiden ongelmista.
- UsableNet, on maksullinen palvelu. Sillä voi tuottaa varsin yksityiskohtaisen raportin koko sivustosta. Palvelussa on maksuttoman kokeilun mahdollisuus.
- Tekstin kieliasun tarkistin, esim. MS Word – ohjelman sisältämä.

Kannattaa kuitenkin muistaa, että nämä välineet testaavat vain teknisiä seikkoja, jotka muodostavat vain pienen osan esteettömyydestä tai kuuluvat yleiseen toimivuuteen (Korpela 2003, 5-10).

2.3 Toteutustekniikat

Web Content Accessibility Guidelines (WCAG) 2.0 antaa suosituksiksi seuraavia tekniikoita, joilla sivustosta saadaan esteetön. Nämä jaetaan neljään kategoriaan, johon kuuluvat havaittavuus, käytettävyys, ymmärrettävyys sekä elinvoimaisuus (W3C WAI 2012).

Havaittavuutta voidaan parantaa esim.:

- Tarjoamalla tekstivaihtoehtoja sisällölle joka ei ole tekstiä.
- Tarjoamalla tekstitystä ja muita vaihtoehtoja multimedialle.
- Luomalla sisältöä, joka voidaan esittää eritavoilla, ilman että sisältö kärsii. Mukaan luettuna auttavat tekniikat.
- Tekemällä käyttäjille helpoksi nähdä ja kuulla sisältö.

Käytettävyyttä voidaan tukea:

- Tekemällä kaikki toiminnot näppäimistöltä saavutettaviksi.

- Antamalla käyttäjille aikaa lukea ja käyttää sisältöä.
- Olemalla käyttämättä sisältöä joka voi aiheuttaa kohtauksia.
- Auttamalla käyttäjiä navigoimaan ja löytämään sisältöä.

Ymmärrettävyyttä voidaan lisätä:

- Tekemällä tekstistä ymmärrettävää ja luettavaa.
- Järjestämällä sisällön ilmestymään ja käyttäytymään ennustettavilla tavoilla.
- Auttamalla käyttäjiä välttämään ja korjaamaan virheitä.

Elinvoimaisuutta voidaan tukea maksimoimalla yhteensopivuus nykyisten ja tulevaisuuden käyttäjätyökalujen kanssa(W3C WAI 2012).

3 TOIMINNALLISET VAATIMUKSET

Tarkoituksena oli luoda sivusto, joka toimii esteettömästi, ja käyttää Wordpressiä vain sisällönhallintaan. Samalla saatiin aikaan toinen sivusto, joka toimii varasivustona, jos ongelmia ilmenee.

Ajatuksena oli luoda moderni sivusto joka käyttää uusia standardeja, eli HTML 5 ja CSS 3, apunaan. Koska sivuston sisältö on luotu dynaamisesti, apuna käytettiin PHP kyselyitä joiden vastaukset tuotiin käyttäjälle näkyviin JavaScriptillä.

Sivuston tuli olla myös luettavissa erilaisilla lukuohjelmilla ja objektien koon ja värityksen sopivat ihmisille joilla on heikentynyt näkö tai värisokeutta. Fonttien koko tuli olla muutettavissa ja sivuston tuli myös toimia mobiililaitteilla.

4 KEHITYSYMPÄRISTÖ

Varsinaisessa ohjelmoinnissa käytin työkaluna Sublime Text 2 kehitysympäristöä. Ohjelma toi hyvin esiin koodissa olevat virheet värikoodauksen avulla ja salli erilaisten formaattien hyvän käytön tallennuksessa.

Sublime Text 2 sisälsi myös kuvan oikealla laidalla sijaitsevan graafisen navigoinnin, jonka avulla pitkässäkin koodissa pystyi nopeasti siirtymään uuteen kohtaan.

Palvelimeen yhteys otettiin graafisella WinSCP FTP sovelluksella, jolla pystyi koodia helposti lataamaan. Graafisuus ei ollut välttämätöntä, mutta käyttömiellyttävyyden vuoksi päädyin siihen.

Wordpressin Codex oli myös jatkuvassa käytössä, koska sieltä löytyivät funktiot ja kirjastot, joita työssä tarvittiin.

Wordpress Codex on online ohjekirja Wordpressille ja elävä tietovarasto Wordpress informaatiolle ja dokumentaatiolle (Wordpress Codex 2014).

5 VERKKOSIVUJEN RESPONSIIVISUUS

5.1 Määritelmä

Tämän päivän tarpeisiin vastaamisen sijaan web sovellukset sovitetaan yleisimpään näytön resoluutioon yhdessä erillisen mobiiliversion kanssa (yleensä vielä tietylle laitteelle suunnitellun). Responsiivisuudessa käännetään idea toisinpäin ja käytetään joustavia ja sulavia sommitteluja, jotka mukautuvat melkein kaikille näytöille (De Graeve 2011).

Responsiivisen Web Suunnittelun (RWD) kolme ominaispiirrettä ovat:

- Mediakyselyt ja mediakyselyiden kuuntelijat.
- Joustava ruudukkopohjainen asettelu, joka käyttää suhteellista koon mitoittamista.
- Mukautuvat kuvat ja media, dynaamisen koonmuutoksen tai CSS avulla.

Todella responsiivisessa websuunnittelussa kaikki kolme kohtaa pitää toteutua. Pääseikka on mukautua käyttäjän tarpeisiin ja laitteiden mahdollisuuksiin (De Graeve 2011).

5.2 Bootstrap

Bootstrap on front – end työkalupaketti nopeasti kehittyville Web sovelluksille. Se on kokoelma yleisiä CSS ja HTML käytäntöjä. Bootstrap käyttää joitakin viimeisimmistä selaintekniikoista tarjotakseen sinulle tyylikkää typografiaa, lomakkeet, napit, taulukot, ruudukot, navigaation ja paljon muuta (Otto 2011).

Projektissa käytettiin apuna Bootstrapin CSS ja JavaScript mallineita, jotka ladattiin index.html-tiedostossa täten tehden sivustosta responsiivisen.

5.3 Viewport meta

Tietyt selaintyypit yrittävät näyttää työpöytä sivustoja pienillä näytöillä automaattisesti zoomaamalla näyttöä. Tämä saattaa olla ongelmallista sovelluksilla, jotka ovat optimoitu pienille näytöille. Viewport meta tagi kertoo laittelle mihin skaalaan sivu tulee piirtää (W3C: Mobile Web Application Best Practices 2014).

Tyypillinen viewport asetus näyttää tältä, `<meta name="viewport" content="width=device-width, initial-scale=1.0"/>`, ja se sijoitetaan HTML dokumentissa `<head>` elementtien sisään. Tämä asetus määrää selainta aina piirtämään sivun 100 % koossa ilman selaimen perustuvaa skaalausta (W3C: Mobile Web Application Best Practices 2014).

Projektissa noudatettiin suositeltua hyvää käytäntöä ja siksi sivusto määriteltiin Viewport meta määritteellä skaalaamaan aina katselussa käytettävän laitteen näyttöön maksimileveyden ollessa laitteen näytön leveys ja aloituskaalan arvon ollessa 1 ja maksimiskaalan arvon ollessa 1 (Liite 1).

5.4 Rivinpituuden responsiivisuudesta

Kaikkea aina 45 merkistä 75 merkkiin saakka pidetään tyydyttävänä rivin pituutena yksipalstaisella sivulla. Yleisesti ideaalisena pidetään 66-merkin rivin pituutta (mukaan lukien kirjaimet ja välilyönnit). Monipalstaiselle työlle parempi keskipituus on 40–50 merkkiä (Webtypography: Choose a comfortable measure 2014).

Projektissa sivuston rivin leveys määriteltiin em -yksikön avulla, ja yhden kappaleen enimmäisleveydeksi määriteltiin 45em:ä. Projektissa tähän 45em arvoon päädyttiin testaamalla useita eri vaihtoehtoja ja tällä arvolla teksti pysyy luettavan levyisenä kuitenkin samalla sivuston tyyliä rikkomatta.

1em on yhtäläinen samanaikaisen fonttikoon kanssa. 2em tarkoittaa siten 2 kertaa nykyisen fontin kokoa. Esim. jos elementti näytetään 12 kuvapisteen fontilla, silloin 2em suuruus on 24 kuvapistettä. Em on hyvin käytännöllinen yksikkö CSS:ssä, koska se sopeutuu automaattisesti fonttiin, jota lukija käyttää (W3School: CSS Units 2014).

Fontin peruskooksi valittiin 16 pistettä, joten edellisen mukaan laskettuna kappaleen enimmäispituudeksi tulee $45 \times 16 = 720$ pistettä. Koska aiemmin mainitun mukaan kappaleen pituus em:nä on riippuvainen fontin koosta ja fontin koko määräytyy kuvapisteinä, teksti skaalautuu hyvin myös pienemmille näytöille.

5.5 Kuvankoon responsiivisuus

Kuvankoon responsiivisuudessa lähdin ajatuksesta, joka jo kohdassa 5.1 mainittiin. Kuvan koko on määritelty näkymään normaalina isoilla näytöillä, mutta @media säännön avulla lisättiin CSS määrittely kuvien leveydeksi kokoruudun leveys ja kuvan korkeus automaattiseksi. Tällä määritelmällä kuvasuhde pysyy vakiona ja samanlaiset määritteet tehtiin myös iframe sisällölle (Liite 7).

Navigointipalkin koko määriteltiin myös @median avulla. Samoin määriteltiin haitarimenun sijainti keskelle ruutua mobiilinäytöissä.

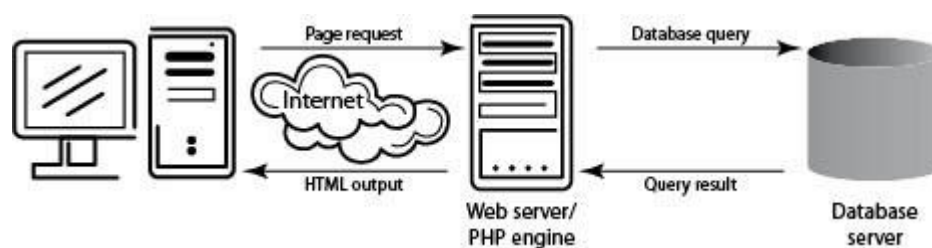
6 VERKKOSIVUJEN DYNAAMISUUS

6.1 Määritelmä

Kun normaalisti web-sivu luodaan HTML ja CSS avulla, kaikki sivuston sisältö on webmasterin kiinteästi asettamaa. Kaikki sivulla kävijät näkevät saman sisällön, sivu on täten staattinen (Powers 2010).

Sitä vastoin dynaamisen websivun sisältö vaihtelee toistuvasti. Esim. uutissivusto Check Magazine:lle näyttää kaksi uusinta artikkeliaan tietokannasta. Kun uusi artikkeli lisätään tietokantaan, PHP koodi näyttää sen automaattisesti. Arkistolinkkiä klikattaessa sisältö vaihtuu näyttämään kyseisen kuukauden artikkelit. Koodi sivustolla kontrolloi sisältöä riippuen selaimelta saapuvista pyynnöistä (Powers 2010).

Kun web-palvelin saa pyynnön, se ohjaa kutsun PHP moottorille, joka normaalisti toimii moduulina palvelimella. Koodista ja pyynnön tyypistä riippuen, PHP moottori lähettää kyselyjä tietokannalle jos se on tarpeellista ja rakentaa sitten HTML tulosteen selaimelle takaisin lähetettäväksi (Powers 2010). Kuva 1 hahmottaa miten web-sivun dynaaminen muodostus toimii.



Kuva 1. Sivun dynaaminen muodostus (Powers 2010).

6.2 Käytännön toiminta

Käytännössä edellisessä kappaleessa mainittu tarkoittaa seuraavaa, sivuston käyttäjä avaa sivun selaimeensa, joka lataa index.html tiedoston. Tiedostossa kysellään bootstrapin JavaScript ja CSS tiedostoja, jQuery kirjastoa, sekä omia tyyli- ja JavaScripttiedostoja, jotka olen tätä työtä varten tehnyt.

Samalla tarkistetaan selainversio, jota käyttäjä käyttää ja ilmoitetaan, jos selain on liian vanha, jotta sivusto toimisi sillä. Mikäli selain on liian vanha, käyttäjän selaimeen ladataan disclaimer.html, jossa kerrotaan, miksi sivu ei näy ja annetaan ohjeistus tilanteen korjaamiseksi.

Index.html luo sivuston rungon, johon ulkonäköä (kuvat 5 ja 6) alkaa muodostaa style.css, style.css lataa myös haitarimenun ulkonäön, joka on sijoitettu erilliseen menu.css tiedostoon pituutensa vuoksi. Samaan aikaan script.js lähettää kyselyitä koskien tiettyjä kategorioita eteenpäin backend_wp.php:lle.

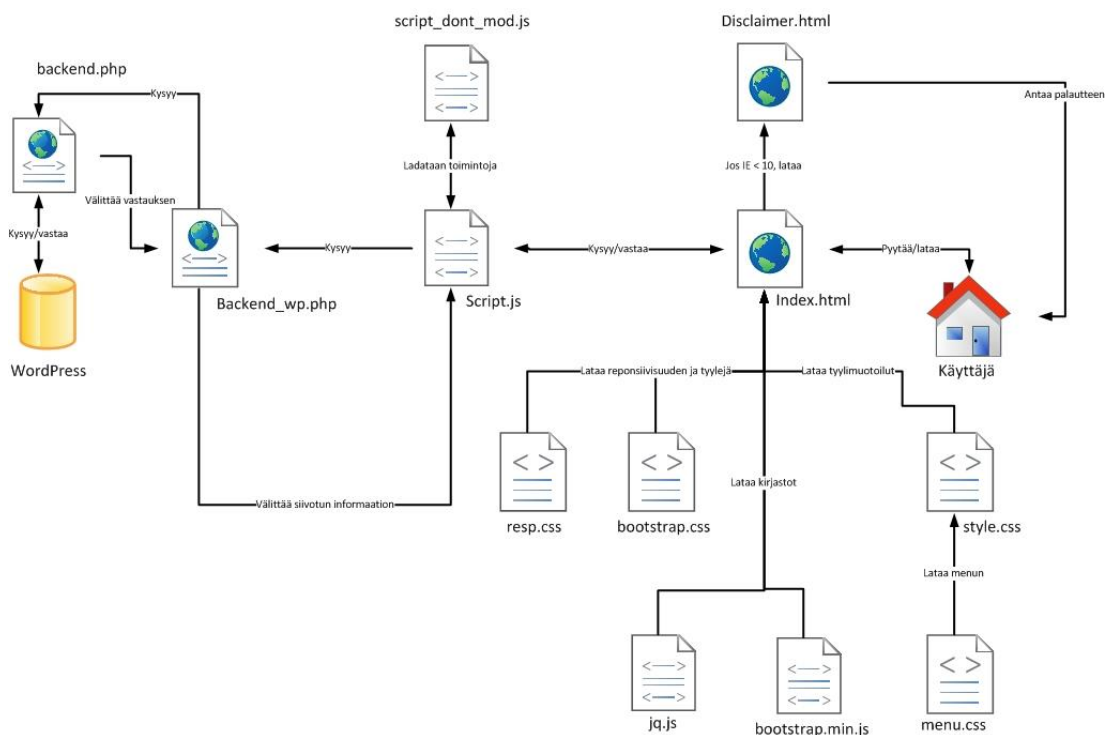
Script.js suorittaa myös funktiokutsuja edelleen script_dont_mod.js tiedostolle, jonka tehtävänä on ladata sivuston toiminnot. Backend_wp.php jaottelee kyselyn omiin alikyselyihinsä ja välittää ne koskien kutakin kategoriaa erikseen backend.php:lle.

Backend.php lataa Wordpressin kirjastot omaan käyttöönsä ja samalla se ottaa vastaan alikyselyt backend_wp.php:lta. Jokaiseen alikyselyn kategoriaan on oma funktionsa joka hakee tiedot Wordpressin tietokannasta ja palauttaa ne takaisin backend_wp.php:lle.

Tämä tiedon haku tehdään kahdessa eri vaiheessa, sillä tiedot palaavat echo:na Wordpressin funktioilta ja näin ne saadaan purettua. Samalla saadaan aikaan selkeyttä.

Backend_wp.php lähettää tiedot eteenpäin takaisin script.js tiedostolle JSON muodossa. Näin haetusta datasta saadaan kasattua siisti paketti, jota voidaan käsitellä helposti myöhemmin.

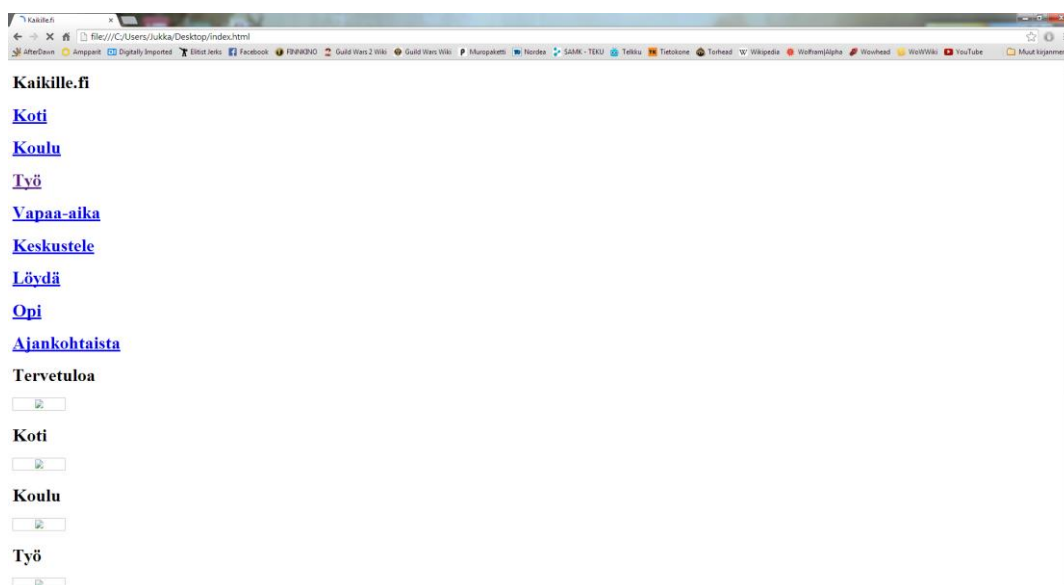
Script.js kasaa saadusta JSON datasta sivuston sisällön, samalla tarkistetaan mm. haitarimenun sisältö. Mikäli haitarimenun ylimmäntason alta löytyy valikoita, se piirittää näytölle. Muussa tapauksessa haitarimenua ei luoda. Bootstrap.css, resp.css ja bootstrap.min.js, vastaavat sivuston responsiivisuudesta. Kuvassa 2 on prosessin havainnollistamiseksi laatimani tietovuokaavio.



Kuva 2. Sivuston www.kaikille.fi tietovuokaavio.

6.3 Index.html

Index.html on sivuston runko joka ei sisällä valmiina mitään elementtejä, joita loppukäyttäjä myöhemmin näkee. Kuvasta 3 käy ilmi, että Index.html sisältää päämenun ja kategoriat joihin myöhemmin haettava data sijoitetaan, kukin kategoriansa mukaan.



Kuva 3. index.html tiedoston muodostama sivuston runko.

Rungon koodi (Liite 1) voidaan jakaa neljään eri osaan ja dokumentin tyyppi on valittu `<!DOCTYPE html>`, joka kertoo selaimelle kyseessä olevan html 5 koodia.

Ensimmäiseksi ladataan meta-määritteet kuten UTF-8, viewport, IE-Edge ja kerrotaan dokumentin sisällön olevan dynaamista. Samalla ladataan Bootstrapin CSS ja JS kirjastot, ladataan jQuery kirjasto, style.css, script.js, script_dont_mod.js sekä ladataan fonttityyppi Googlen API:sta.

Toisessa kohdassa suoritetaan Google Analytics alustus ja suoritus kävijätilastointia varten sekä selaintarkistus.

Kolmanneksi kohdaksi voidaan jakaa kohtaan, jossa muodostetaan varsinainen navigointivalikko sivuston yläreunaan. Neljännessä kohdassa suoritetaan varsinaisten sisältöosioiden luominen ja ladataan ennen sisällön näyttöä loader.gif osoittamaan sivustolla kävijöille, että sivua ladataan.

6.4 Disclaimer.html

Tämän tiedoston muodostama sivu ladataan, joka esiintyy kuvassa 4, selaimen epäonnistuessa läpäisemään versiotarkistuksen. Kyseessä on hyvin yksinkertainen html-sivu (Liite 2) joka ilmoittaa selaimen olevan liian vanha, ja joka antaa käyttäjälle vaihtoehtoja uuden selaimen hankkimiseksi.

Kaikki sivustolla sijaitsevat linkit vievät selainvalmistajien sivun kohtaan josta löytyy aina uusin versio. Selaimet joita sivulla opastetaan käyttämään, ovat uusin Internet Explorer, Chrome tai Firefox. Sivulla myös kehoitetaan organisaatioissa työskenteleviä ottamaan yhteyttä tietohallintoon.



Kuva 4. Opastus sivusto.

6.5 Script.js

Script.js tiedostoa kutsutaan index.html tiedostossa. Kutsu toteutetaan script tagien sisällä src-nimisen attribuutin kertoessa mistä kyseinen tiedosto löytyy (Liite 1).

Script.js tiedosto jakautuu kolmeen erilliseen osaan. Ensimmäisessä osuudessa käydään läpi elem_ids nimisen taulukon sisältö, joka muodostuu edeltä määritellyistä kategorioista ja niiden WordPress ID numeroista (Liite 3). Haluttujen kategorioiden löytämiseksi kyseisiä numeroita ei saa muuttaa eikä sivuja tuhota sillä WordPress asettaa ne satunnaisesti.

Tämä elem_ids taulukko ajetaan suoraan sisälle loadContent funktiolle, joka ottaa tiedot vastaan järjestyksessä wanted_id ja class_id. Wanted_id on edellä mainittu WordPress:n antama tunnus numero muodossa, class_id taas on elementin nimi.

Saatuinaan nämä tiedot loadContent funktio lähettää Ajaxilla GET-kutsun backend_wp.php tiedostolle, GET-kutsu sisältää ohjeistuksena wanted_id:n ja class_id:n. GET-kutsun cache ominaisuuden arvoksi on määritelty false, sillä GET-kutsujen tuloksia tallennus välimuistiin dynaamisilla websivustoilla ei ole haluttua sivuston sisällön ollessa muuttuvaa (Api.JQuery 2014).

Tämän jälkeen GET-kutsun hakema informaatio tallennetaan data-nimiseen muuttu-
jaan ja se käsitellään `parseJSON(data)` komennolla. Käsittelyn lopputulos tallennetaan
takaisin data-nimiseen muuttujaan, tämä korvaa muuttujan alkuperäisen sisällön. Ko-
mento `parseJSON()` muodostaa JSON-taulukosta JavaScriptin luettavan merkkijonon
(Api.JQuery 2014).

Funktion alussa `class_id`:stä rakennetaan myös kaksi erillistä muuttujaa, `$el_id` ja
`$el_class` myöhempää elementtien käsittelyä helpottamaan. Varsinkin `$el_id`:tä käyte-
tään tunnistamaan eri kategorioiden sisälle rakennettuja sisäkkäisiä elementtejä. Esi-
merkkinä tästä toimii kaikkien kategorioiden ”.content-header” elementti. Elementin
sisältö etsitään mm. `<h1>` headeria, jonka sisältö tyhjennetään ja sinne tallennetaan
featured image tai thumbnail riippuen siitä kumpaa termiä haluaa käyttää.

Tämän jälkeen suoritetaan data muuttujaan tallennettujen otsikon, sisällön, komment-
tien, menun sekä data-muuttujasta riippumattoman koko elementin tietojen haku ja
kotelointi html elementeillä. Samalla nämä uudet elementit tietoineen tallennetaan
omiin muuttujiinsa.

Seuraavaksi kyseiset elementit sijoitetaan `append_this` nimiseen muuttujaan, jota täy-
dennetään html elementeillä. Tätä edelleen laajennetaan myöhemmin etsimällä otsi-
kolle ja koko elementille oikeat paikkansa sisäkkäisistä html elementeistä ja täyden-
netään paikoilleen `append_this` muuttujan sisällöllä.

Samalla myös tarkistetaan menu elementin koko. Jos menu sisältää enemmän kuin 1
portaan eli äiti – tytär hierarkian äidin, sivuston kyseiseen kohtaan sijoitetaan menu
elementti.

Tämän jälkeen suoritetaan `dynamicUI()`, joka vastaa kommenttiosion näyttämisestä.
Tätä seuraa `formFunction(wanted_id, class_id)` toiminto, joka vastaa menun perus-
teella halutun sisällön lataamisesta oikeaan kategoriaansa. Funktiot `textSize()` ja `ac-
cordionMenu()` luovat tekstinkoon vaihtoon tarvittavan objektin ja sivuston selailuun
liittyvän menun.

Funktio nimeltä `formFunction()` toimii samalla periaatteella, kuin edellinen `loadContent()` funktio mutta keskittyy vain yhden kategorian sisällön ja menun käskyjen tarkkailuun.

Haitari eli accordion menun avulla suunnistus on toteutettu tässä tarkkailemalla käyttäjän painalluksia menussa. Menua painettaessa selvitetään linkki jota listassa painettiin. Seuraavaksi erotetaan linkin html-elementeistä varsinainen ladattavaksi halutun sivun webosoite. Pelkkää osoitetta on helpompi käsitellä ja se jaetaan kahtia kohdasta `”page_id=”`.

Jälkimmäinen puolisko, joka on numero, otetaan talteen `linkHref` muuttujaan ja käytetään yksittäisen sivun latauksessa. Muuttujaa `linkHref` käytetään vastaavanlaisesti ajaxin GET-kutsussa kuin `wanted_id`:tä. Tämän jälkeen kategorian, jonka haitarimenua käytettiin, sisältö pyyhitään tyhjäksi ja täytetään menun avulla kutsutun sivun sisältämällä informaatiolla.

6.6 Script_dont_mod.js

Tiedoston `script_dont_mod.js` tarkoituksena on toimia varastona funktioille, joita kutsutaan suoritettaessa `script.js` tiedoston sisältöä (Liite 4). Kuten `script.js` tiedoston tapauksessakin, `script_dont_mod.js` tiedoston sijainti ja käyttöönotto tapahtuu `index.html:ssä`.

Ensimmäisenä on `UIfunctions()` niminen toiminto, joka vastaa navigointi valikon tarkkailusta ja sivuston vierittämisestä haluttuun kohtaan. Tämä funktio myös vastaa navigaatiopalkin tilan muuttamisesta lisäämällä tai poistamalla luokan `active` sen mukaan kuinka alas sivustolla on menty. Tämän tiedon perusteella `style.css` määrittelee navigointipalkin ulkonäön.

`DynamicUI()` funktio taas tarkkailee kommenttiosion tilaa. Kommentoi nappia painettaessa tarkistetaan nappia seuraavan luokan nimi ja lisätään tai poistetaan luokka `open` tilasta riippuen, myös nimeä vaihdetaan elementille riippuen siitä onko kommentointi osio auki vai kiinni.

Menun toiminnasta vastaava `accordionMenu()` funktio taas puolestaan määrittelee menun aina ladattaessa muotoon jossa päämenun alaiset lapsimenut ovat suljettu, tämä pätee myös lapsenlapsi menuihin. Samalla myös määritellään menujen pysyminen auki kun niissä navigoidaan.

Seuraavana `accordionMenu()`:ssa on vuorossa mitä tehdään kun menua klikataan, ensin tarkistetaan onko jonkin menuelementin tila näkyvillä vai ei. Tilan ollessa muuta kuin `”:visible”` menu elementin alaiset lapsielementit tuodaan esille. Samalla niiden piirtonopeus määritellään nopeaksi.

Kaikki muut avonaiset menut suljetaan samalla tilan säästämiseksi. Menun varsinainen ulkoasu määrytyy `menu.css` tiedostossa olevasta tyylimäärittelystä (Liite 8). Samassa tiedostossa määritellään myös efektit jota menu käyttää tarkemmin.

Tekstin koon muuttamisesta vastaa funktio nimeltä `textSize()`, jossa aluksi tehdään koodin selkeyden vuoksi kaksi muuttujaa `$normal` ja `$large`. Näihin muuttujiin tallennetaan tieto tekstin muuttamiseen käytettyjen elementtien `id`:stä.

Normaalitekstin kooksi on määritelty kappaleissa ja listaelementeissä 16 pikseliä, kun otsikoiden koko on taasen 36 pikseliä. Suurennetun tekstin koko on kappaleissa ja listaelementeissä 20 pikseliä, kun otsikon koko on taas 45 pikseliä.

Viimeisenä on selaimen tyypin ja version tarkistukseen käytettävä funktio nimeltä `getInternetExplorerVersion()`, jonka tarkoituksena on selvittää voiko sivustoa käyttää käyttäjän käyttämällä selaimella. Mikäli selain läpäisee tarkistuksen, käyttäjälle näytetään `index.html` tiedoston rungon pohjalle rakennettu sivusto. Muussa tapauksessa ruudulle ladataan `disclaimer.html` ohjeistusta varten.

Tarkistus toimii seuraavasti, ensin tarkastetaan selainsovelluksen nimi, jonka jälkeen suoritetaan versionumeron tarkistus. Selainsovelluksen ja version ollessa tiedossa nämä tiedot palautetaan `index.html` tiedostolle, joka kutsui funktiota. Tämän jälkeen `index.html` tiedostossa ajetaan JavaScript koodi, joka selvittää palautetun tiedon sisällön ja päättää sen perusteella ladataanko sivusto vaiko `disclaimer.html`.

Tarkistuksen läpäisyyn soveltuvia selaimia ovat Internet Explorer 10.0 ja sitä myöhemmät selainversiot. Muut selaimet tukevat sivustoa käytännössä todella vanhoihin versioihin saakka joten niiden kanssa ei törmätä ongelmiin.

6.7 Backend_wp.php

Tiedosto backend_wp.php ottaa vastaan kyselyitä script.js tiedostolta wanted_id muuttujaan tallennetun WordPress:n oman sisäisen ID numeron perusteella (Liite 5). Selkeyden vuoksi backend_wp.php tiedoston saama informaatio tallennetaan muuttujaan \$id.

Seuraavaksi luodaan viisi muuttujaa, joihin tallennetaan informaatio joka saadaan backend.php tiedostolle lähetettyjen GET-kyselyiden perusteella. Haluttuja tietoja ovat sivuston sisältö, kommentit, otsikot, thumbnail – kuvat sekä menut.

Sitten luodaan JSON taulukko, joka on muodossa nimi/arvo. Esimerkkinä tällaisesta on mm. "firstName":"John" (W3School: JSON Syntax 2014). Taulukkoon tallennettavien muuttujien pareiksi on tässä tapauksessa laitettu nimet jotka vastaavat muuttujan nimeä ja kuvaavat muuttujan sisältöä.

JSON taulukko palautetaan takaisin kutsun esittäneelle script.js tiedostolle, joka osaa määrittää JSON taulukon nimien perusteella mitä tietoa data muuttujasta erityisesti halutaan. Esimerkkinä toimikoon JSON taulukon pari "thumbnail" => \$thumbnail, joka sallii kysyä data muuttujasta suoraan kyselyllä data.thumbnail vain pelkästään thumbnail kuvaa.

Tärkein tehtävä backend_wp.php tiedostolla on selkeyttää tiedonkulkua backend.php ja script.js tiedostojen välillä ja muodostaa haetusta datasta siisti nippu jota on helpompi käsitellä myöhemmin. Samalla saadaan poistettua WordPress funktioiden omien funktioiden echotusten aihettamat ongelmat. Echotus on toiminto, joka tulostaa yhden tai useamman merkkijonon tai vastaavasti kaikkien saamiensa parametrien sisällön (Php.net: Echo 2014).

Laatimalla tämä tiedosto saatiin aikaan myös yllättävänä positiivisena sivuvaikutuksena sivustolle sietokykyä mahdollisia muutoksia kohtaa. Tämä tuli esiin testipalvelinta rakennettaessa, jonne sivusto saatiin laitettua minimaalisilla muutoksilla. Nämä muutokset kohdistuvat vain ja ainoastaan backend_wp.php tiedostoon. Muutosten jälkeen testipalvelin rupesi toimimaan erittäin nopeasti.

6.8 Backend.php

Tiedosto backend.php toimii rajapintana WordPressin ja muun koodin välillä. Tiedoston alussa ladataan WordPressistä kirjastoja muistiin. Näiden kirjastojen avulla on mahdollista hakea funktioita ohjeistamalla informaatiota WordPressin omasta tietokannasta jonne sivujen yms. tiedot päätyvät niitä luotaessa (Liite 6).

Kirjastojen jälkeen seuraa ehtolauseita, jotka tarkkailevat tulevien GET-kutsujen tyyppiä. Tyypin mukaan päätetään minkä ehtolauseen sisäinen koodi suoritetaan. Tarkkailtavia kutsuja ovat mm. get_thumbnails, get_titles, get_all_content, get_comments sekä get_menu.

Kaikkien ehtolauseiden sisällä on myös The Loop nimellä tunnettu silmukka, jota tarvitaan tiettyjen funktioiden toimimiseksi. Samalla kaikkiin ehtoihin on sisällytetty taulukko, joka listaa argumentit joiden perusteella The Loop käy sivuja ja informaatiota läpi (WordPress: Codex The Loop 2014).

Vasta The Loopin sisälle tulee ehtolauseen suoritettava koodi, joka suodattaa The Loopin hakemasta datasta halutun informaation ja suorittaa informaation echotuksen. Tämä echotettu data jälleen käsitellään puolestaan backend_wp.php tiedostossa. Jokainen ehtolause sisältää myös ilmoituksen siitä, että viestejä ei löydy jos The Loopin haussa ei dataa löydy.

Menu elementtejä käsittelevässä ehtolauseessa tehdään edellä olevien asioiden lisäksi taulukko nimeltä \$args, joka listaa kaikkien päätason alaiset eli äiti – lapsi hierarkian lapsi ja lapsenlapsi sivut.

Näiden sisältö käydään samalla läpi yksitellen ja niistä rakennetaan \$pageIds taulukko, jonka avulla rakennetaan argumentti taulukko \$args. Taulukko annetaan funktion wp_page_menu() käsiteltäväksi ja se rakentaa karkean listan, jossa päämenun alla on alimenuja ja niiden alaiset menut sikäli kun niitä löytyy. Näin valmistuu karkea menu joka saa lopullisen ulkoasunsa menu.css avulla ja jonka toimintoja ohjataan script_dont_mod.js sisältävän funktion avulla.

6.9 Style.css

Sivuston ulkoisesta tyylistä ja värimaailmasta aina päävalikon nappien muotoiluun vastaa style.css tiedosto. Tässä tiedostossa on määritelty ns. nested- eli sisäkkäisten elementtien avulla eri alueiden tyylit, joten tiedostosta tulee pakostakin pitkä. Kuvassa 5 näytetään sivuston ulkoasu, joka on rakennettu style.css ja muiden edellä mainittujen tiedostojen yhteisvaikutuksen ansiosta.

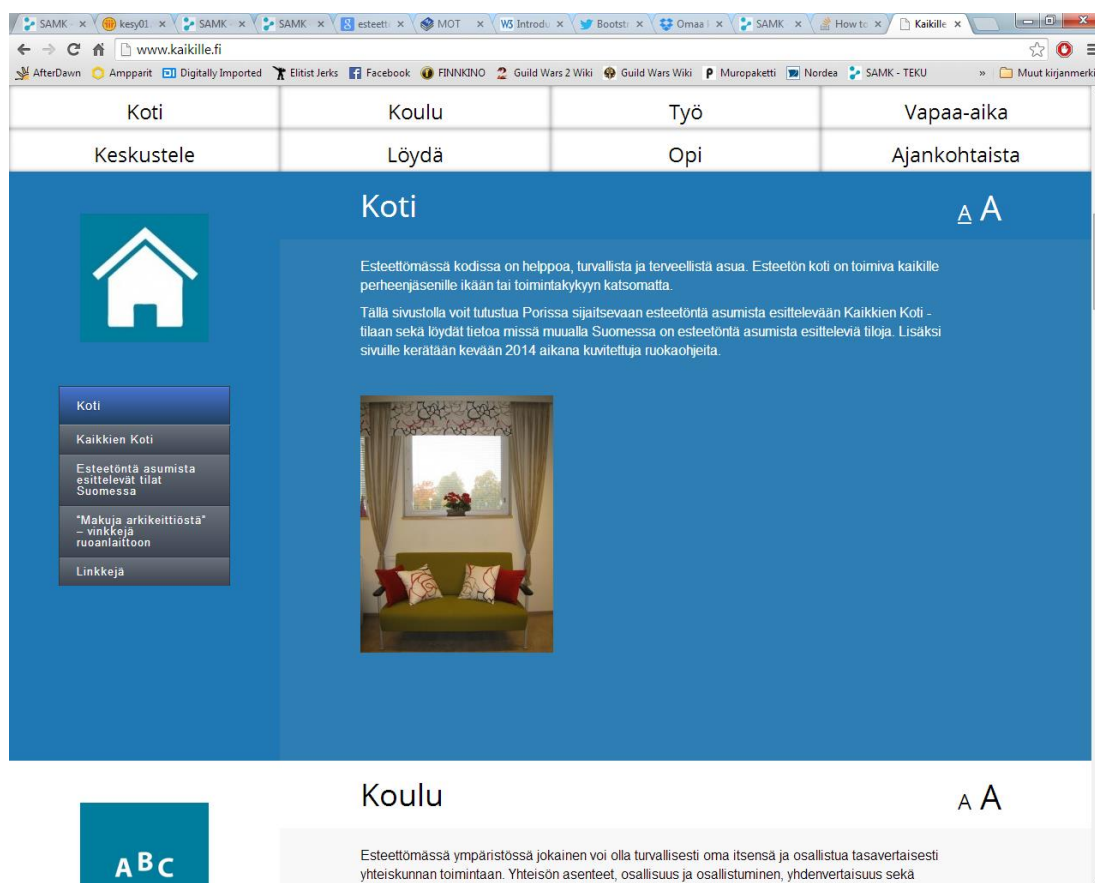


Kuva 5. Sivuston ulkoasu, kun sivu ladataan.

Esimerkkinä tästä sisäkkäisyydestä toimivat `.header` ja `.header h1`, jossa toinen määrittää `.header` luokan sisällön keskitettäväksi ja toinen antaa määrittelyt `h1` otsikko elementin tekstin koolle ja rivinkorkeudelle luokan `.header` sisällä.

Tiedosto `style.css` määrittelee myös sivuston tulostamisen mobiilinäyttöillä `@media` (`max-width: 640px`) kyselyllä. Mobiilinäyttöillä kuvien kokoa myös kutistetaan siten, että kuvasuhde pysyy samana. Tämä on toteutettu sallimalla maksimileveydeksi elementin sisäinen maksimi johon kuva piirretään ja korkeus määritellään automaattiseksi.

Kuvasta 6 näkyy navigoinnin ulkonäkö kun sivustoa vieritetään riittävän alas, tällöin navigointi kutistuu ylös palkiksi eikä häiritse sisällön katselua. Myös tämän navigoinnin efektit ja ulkoasu on määritelty `style.css` tiedostossa.



Kuva 6. Sivusto ja navigointi, kun sivua vieritetään alaspäin riittävästi.

Sivuston efektit ja värimaailma on pidetty yhtenäisenä. Kun menun tai navigoinnin yllä pidetään hiirtä ne muuttuvat oransseiksi. Oranssi väritys eroaa selkeästi muusta värityksestä ja sen huomaa selvästi. Navigoinnin ja tekstin koossa on huomioitu helpolukuisuus. Linkkien väritys on myös oranssi yhtenäisyyden ja erottuvuuden vuoksi, sillä linkkien tulee näkyä myös siniseltä pohjalta.

6.10 Menu.css

Menu elementille, joka esiintyy kuvassa 6, värimaailmaksi valittiin harmaan eri sävyjä erottelemaan eri tasot toisistaan. Samalla erotellaan menu omaksi elementikseen erikseen muista sivulla esiintyvistä elementeistä.

Menun tekstin väritykseksi valittiin valkoinen paremman näkyvyyden vuoksi. Hiiren osoitinta leijuttamalla menun jonkin kohdan yllä kyseinen kohta värjätään oranssiksi ja teksti vaihdetaan mustaksi.

Menun tyylit jouduttiin rakentamaan sisäkkäisten elementtien vuoksi joka tasolle erikseen. Tapa jolla haitarimenu rakennetaan johtaa useisiin sisäkkäisiin elementteihin ja tästä johtuen tyylitiedoston koodista tulee väkisin pitkä.

7 TESTAUS

7.1 Verkkosivujen esteettömyyden ja käytettävyyden arviointi

Sivuston käytettävyyden ja esteettömyyden arviointi perustuu WAI:n (W3C) julkaisemiin suosituksiin verkkopalveluiden esteettömyydestä. Tämän lisäksi toteutimme kaksi verkkokyselyä, joiden kohderyhmänä olivat näkövammaiset ja kehitysvammaiset. Kyselyt koostuivat muutamista tehtävistä sekä monivalintakysymyksistä, joissa kartoitettiin esteettömyyden ohella myös käytettävyyttä ja käyttäjäkokemusta.

Molempiin käyttäjäkyselyihin vastasi vain pieni joukko käyttäjiä. Toisaalta vastauksissa ei ollut kovinkaan suurta hajontaa ja molemmilla kerroilla osa vastauksista oli hyvin samansuuntaisia. Tutkimuksen vastaajiksi kutsuttiin henkilöitä useista eri vammaisryhmistä, mutta valtaosaltaan vastaajat olivat näkövammaisia ja miehiä. Ensimmäisellä kerralla vastaajia oli 10 ja toisella kerralla 11 (Karinharju, Kuusinen, Tupala, Vuorinen, Hammais, Virtanen & Halttunen 2013, 6).

Sivuston teknisen tehokkuuden arviointi on laitekannan kirjavuuden takia haastavaa, mutta tehtyjen käytettävyyystestausten perusteella kaikille.fi sivusto toimii erilaisilla verkkoselaimilla ja päätelaitteilla. Käytettävyydestä vastanneista kukaan ei käyttänyt mobiililaitteita.

Kyselyyn vastanneet käyttäjät antoivat myönteistä palautetta erityisesti sivuston käyttöliittymästä ja sen toimivuudesta erilaisilla apuvälineillä. Syy ei kuitenkaan välttämättä ole itse sivustossa, vaan apuvälineiden, kuten puhesynteesiohjelmien toimivuudessa.

Sivuston käyttöliittymä koettiin pääsääntöisesti toimivaksi. Käyttöliittymän puutteiksi koettiin pitkät ja monipolviset alas vetolaatikot, joiden käyttäminen vaatii hyviä motorisia taitoja (Karinharju, Kuusinen, Tupala, Vuorinen, Hammais, Virtanen & Halttunen 2013, 7).

7.2 Kaikille.fi-sivuston käytettävyyden kehittäminen ja heuristinen arviointi

Sivustolle suoritettiin käytettävyyssuunnittelijan toimesta heuristinen testaus, joka paljasti sivustolta neljä suurta käytettävyysongelmaa, yhdeksän pientä ongelmaa ja neljä kosmeettista ongelmaa. Esimerkkinä suuresta ongelmasta oli tekstin äänitettyjen versioiden toimimattomuus joillakin selaimilla.

Ongelma havaittiin, syy selvitettiin ja korjaaminen tapahtui käytännössä äänitiedostojen transkoodaamisella esteettömämpään tiedostomuotoon (Karinharju, Kuusinen, Tupala, Vuorinen, Hammais, Virtanen & Halttunen 2013, 7-8).

Testissä paljastui myös useita sivuston hyviä puolia, kuten palvelun tilaan liittyvä palaute, jota käyttäjä saa useimmissa tapauksissa riittävästi. Tällöin käyttäjä tuntee olonsa kotoisaksi ja turvalliseksi tietäen koko ajan, missä kohtaa sivustoa hän on milloinkin.

Navigoinnin käytettävyyttä parannettiin korvaamalla perinteiset alas vetolaatikot täysin uuden tyyppisellä accordion-menulla eli haitarivalikolla. Tällaisessa menu-tyypissä selattava kategoria ja sen alainen sisältö on esillä, mutta sisältö, jota ei tarkastella sulkeutuu palkeiden tapaisesti (Karinharju, Kuusinen, Tupala, Vuorinen, Hammais, Virtanen & Halttunen 2013, 8).

8 PARANNUSEHDOTUKSET

Ensimmäisenä parannuksena sivustolle pitäisi seuraavaksi laatia Google Mapsin JavaScript API v3:een perustuva karttapalvelu, joka sisältää karttamerkintöjä ja info laatikoita erilaisista esteettömistä palveluista. Karttapalvelun merkinnät tulisi voida myös piilottaa näytöltä kokonaan tai vain haluttu kategoria kerrallaan. Karttainformaatio voitaisiin tallentaa vaikkapa WordPressiin, josta se noudetaan kartalle tulostettavaksi.

Koodia voitaisiin myös optimoida nopeammaksi ja paremmin nykyiset mobiilikäytön kriteerit täyttäväksi. Sivuston suoritusnopeutta voitaisiin parantaa huomattavasti minimoimalla suoritettavien tiedostojen koko. Tyyli-tiedostojen optimointiin pitäisi tässä kiinnittää myös erityistä huomiota, sillä siellä voinee hyvinkin olla erilaisia määrittelyitä jotka voitaisiin yhdistää ja näin minimoida tyyli-tiedoston koko.

Nämä CSS-tiedostot voitaisiin vielä muokata mobile first mallin mukaisiksi, sekä myöhemmin CSS 4 standardin ilmestyessä muuttaa kyseiseen standardiin koon pudottamiseksi puoleen sekä nested elementtien paremman hallinnan vuoksi.

Sivuston keskustelu alue odottaa edelleen tekijäänsä ja sivustolla on varmasti jossain piilossa vielä virhe jos toinenkin, joita ei ole kuitenkaan vielä löydetty. Näiden poistamiseen tulee kiinnittää erityistä huomiota. Yksi kehitysmahdollisuus on tutkia Facebookin integrointia sivustoon siten että se toimisi keskustelualustana.

9 YHTEENVETO

Ennen työn aloittamista hieman jo arvelutti, kun kuulin homman olevan valmiiksi jo myöhässä. Hommaa ei myöskään helpottanut tietää että edeltäjäni oli yrittänyt saada tietoja ulos WordPressistä, mutta oli nopeasti luovuttanut asian suhteen.

Tästä huolimatta tartuin työhön ja asetin tavoitteeksi saada ensin jotain ulos WordPressistä. Vasta sen jälkeen voisi asiaa alkaa laajentamaan eteenpäin. Oli haastavaa tutkia tämän julkaisujärjestelmän funktioita, mutta pala kerrallaan onnistuin saamaan asioita toimimaan.

Työ oli sekä haastavaa että mielenkiintoista, sivustosta muotoutui suurin piirtein sellainen kuin oli visioitukin. Sivusto on lakanaluonteensa vuoksi hankalasti muokattava, mutta se onnistuu kyllä. Työ palkitsi, sillä koin tekeväni jotain, josta on myöhemmin iloa ja hyötyä ihmisille. Sivustolle on puuhattu jatkokehitystä juuri karttapalvelun kehityksen tiimoilta.

LÄHTEET

Kuusinen, J & Tupala, R. 2013. Internetsivujen esteettömyys. Teoksessa Jutila, S. ja Ilola, H. (toim.) Matkailua kaikille? Näkökulmia matkailun ennakointiin, osa II. Matkailualan tutkimus- ja koulutusinstituutti. Rovaniemi, 86-89.

Kuusinen, J & Vuorinen, J. 2014. Creation of an Interactive and Accessible Online Learning Environment. Viitattu 30.4.2014. http://www.thinkmind.org/index.php?view=article&articleid=web_2014_1_40_40029

De Graeve, K. 2011. HTML 5 – Responsive Web Design. Viitattu 14.5.2014. <http://msdn.microsoft.com/en-us/magazine/hh653584.aspx>

Otto, M. 2011. Bootstrap from Twitter. Viitattu 14.5.2014. <https://blog.twitter.com/2011/bootstrap-twitter>

W3C: Mobile Web Application Best Practices 2014. Viitattu 14.4.2014. <http://www.w3.org/TR/mwabp/#bp-viewport>

Powers, D. 30.6.2010. Building your first dynamic website – Part 1: Setting up the site and database. Viitattu 14.4.2014. http://www.adobe.com/devnet/dreamweaver/articles/first_dynamic_site_pt1.html

W3School: CSS Units 2014. Viitattu 15.4.2014. http://www.w3schools.com/cssref/css_units.asp

Webtypography: Choose a comfortable measure. Viitattu 16.4.2014. <http://webtypography.net/2.1.2>

W3C WAI: Introduction to Web Accessibility 2005. Viitattu 14.5.2014. <http://www.w3.org/WAI/intro/accessibility.php>

Wordpress: Codex 2014. Viitattu 14.5.2014. https://codex.wordpress.org/Main_Page

Api.JQuery: jQuery.ajax() 2014. Viitattu 27.10.2014. <http://api.jquery.com/jquery.ajax/>

Api.JQuery: jQuery.parseJSON() 2014. Viitattu 27.10.2014. <http://api.jquery.com/jquery.parsejson/>

W3School: JSON Syntax 2014. Viitattu 29.10.2014. http://www.w3schools.com/json/json_syntax.asp

Php.net: Echo 2014. Viitattu 29.10.2014. <http://php.net/manual/en/function.echo.php>

WordPress: Codex The Loop 2014. Viitattu 29.10.2014. http://codex.wordpress.org/The_Loop

Karinharju, K., Kuusinen, J., Tupala, R., Vuorinen, J., Hammais, E., Virtanen, P. & Halttunen, P. 2013. Interaktiivinen ja esteetön verkko-oppimisympäristö kaikille.fi. Viitattu 30.10.2014. <http://www.kaikille.fi/wordpress/wp-content/uploads/2013/11/KAIKILLE.FI-interaktiivinen-verkko-oppimisymp%C3%A4rist%C3%B6-loppuraportti.pdf>

Korpela, J. www-sivut jokaiselle sopiviksi. 2003. Viitattu 6.11.2014. <http://www.tieke.fi/pages/viewpage.action?pageId=15112643>

W3C WAI: WCAG 2 at a Glance. 2012. Viitattu 6.11.2014. <http://www.w3.org/WAI/WCAG20/glance/>

LIITTEET

- LIITE 1 Index.html rungon koodi.
- LIITE 2 Disclaimer.html sivun koodi.
- LIITE 3 Script.js tiedoston koodi.
- LIITE 4 Script_dont_mod.js tiedoston koodi.
- LIITE 5 Backend_wp.php tiedoston koodi.
- LIITE 6 Backend.php tiedoston koodi.
- LIITE 7 Style.css tiedoston koodi.
- LIITE 8 Menu.css tiedoston koodi.


```

<!DOCTYPE html>

<html>
<head>
<title>Kaikille.fi</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-
scale=1">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="document-state" content="dynamic">
<!--<meta http-equiv="Cache-control" content="public"-->
<link href='http://fonts.googleapis.com/css?family=Oriental|Cinzel+Decora-
tive:400,700,900' rel='stylesheet' type='text/css'>
<link href='css/bootstrap.css' rel='stylesheet' type='text/css'>
<link href='css/style.css' rel='stylesheet' type='text/css'>
<link href='css/resp.css' rel='stylesheet' type='text/css'>

<!--JQ-->
<script src='js/jq.js'></script>
<!--BS-->
<script src="js/bootstrap.min.js"></script>
<!--Custom-->
<script src='js/script_dont_mod.js'></script>
<!--Custom-->
<script src='js/script.js'></script>
<!--Custom-->

<!--Google Analyticsin oma koodi-->
<script>
  (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
  (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
  m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
  })(window,document,'script','//www.google-analytics.com/analytics.js','ga');

  ga('create', 'UA-48150829-1', 'kaikille.fi');
  ga('send', 'pageview');

</script>

<!--Browser version check-->
<script>
  var msg = "You're not using Internet Explorer."; //Save message for test pur-
                                                    poses.
  var ver = getInternetExplorerVersion();          //Get IE version.

  if ( ver > -1 )
  {
    if ( ver >= 10.0 ) {
      //In here we used to show message if you were using up to date browser.
      //msg = "You're using a recent copy of Internet Explorer.";
    }
    else{
      //We load disclaimer.html here if the check fails.
      window.location.href = "../disclaimer.html";
    }
  }
</script>

</head>
<!--Initialize navigation scrolling-->
<body data-spy="scroll" data-target="#nav">
  <div class="container">

```

```

<div class="row-outer row-outer-header">
  <div class="row">
    <div class="header col-md-12">
      <div class="header-inner">
        <h1>Kaikille.fi</h1>
      </div>
    </div>
  </div>
</div>

<!--Navigation starts-->
<div id="#nav" class='fix'></div>
<div class='navigation'>
  <div class="row-outer row-outer-nav">
    <div class="row">
      <a href="#home">
        <div class="home nav-header col-md-3">
          <div class='nav-header-inner'>
            <h1>Koti</h1>
          </div>
        </div>
      </a>
      <a href="#school">
        <div class="school nav-header col-md-3">
          <div class='nav-header-inner'>
            <h1>Koulu</h1>
          </div>
        </div>
      </a>
      <a href="#work">
        <div class="work nav-header col-md-3">
          <div class='nav-header-inner'>
            <h1>Työ</h1>
          </div>
        </div>
      </a>
      <a href="#freetime">
        <div class="freetime nav-header col-md-3">
          <div class='nav-header-inner'>
            <h1>Vapaa-aika</h1>
          </div>
        </div>
      </a>
    </div>
  </div>

  <div class="row-outer row-outer-nav2">
    <div class="row">
      <a href="#ask">
        <div class="ask nav-header col-md-3">
          <div class='nav-header-inner'>
            <h1>Keskustele</h1>
          </div>
        </div>
      </a>
      <a href="#find">
        <div class="find nav-header col-md-3">
          <div class='nav-header-inner'>
            <h1>Löydä</h1>
          </div>
        </div>
      </a>
      <a href="#learn">
        <div class="learn nav-header col-md-3">
          <div class='nav-header-inner'>
            <h1>Opi</h1>
          </div>
        </div>
      </a>
    </div>
  </div>

```

```

        </div>
      </div>
    </a>
    <a href="#now">
      <div class="now nav-header col-md-3">
        <div class='nav-header-inner'>
          <h1>Ajankohtaista</h1>
        </div>
      </div>
    </a>
  </div>
</div>

<!--Content starts-->
<div id="intro" class="row-outer row-outer-content">
  <div class="row">
    <div class="content-header col-md-3">
      <h1>Tervetuloa</h1>
    </div>
    <div class="content-text col-md-9">
      <div class='content-inner'>
        <div class='loader'></div>
      </div>
    </div>
  </div>
</div>

<div id="home" class="row-outer row-outer-content">
  <div class="row">
    <div class="content-header col-md-3">
      <h1>Koti</h1>
    </div>
    <div class="content-text col-md-9">
      <div class='content-inner'>
        <div class='loader'></div>
      </div>
    </div>
  </div>
</div>

<div id="school" class="row-outer row-outer-content">
  <div class="row">
    <div class="content-header col-md-3">
      <h1>Koulu</h1>
    </div>
    <div class="content-text col-md-9">
      <div class='content-inner'>
        <div class='loader'></div>
      </div>
    </div>
  </div>
</div>

<div id="work" class="row-outer row-outer-content">
  <div class="row">
    <div class="content-header col-md-3">
      <h1>Työ</h1>
    </div>
    <div class="content-text col-md-9">
      <div class='content-inner'>
        <div class='loader'></div>
      </div>
    </div>
  </div>
</div>

```

```

        </div>
      </div>
    </div>
  </div>

  <div id="freetime" class="row-outer row-outer-content">
    <div class="row">
      <div class="content-header col-md-3">
        <h1>Vapaa-aika</h1>
      </div>
      <div class="content-text col-md-9">
        <div class='content-inner'>
          <div class='loader'></div>
          </div>
        </div>
      </div>
    </div>
  </div>

  <div id="ask" class="row-outer row-outer-content">
    <div class="row">
      <div class="content-header col-md-3">
        <h1>Keskustele</h1>
      </div>
      <div class="content-text col-md-9">
        <div class='content-inner'>
          <div class='loader'></div>
          </div>
        </div>
      </div>
    </div>
  </div>

  <div id="find" class="row-outer row-outer-content">
    <div class="row">
      <div class="content-header col-md-3">
        <h1>Löydä</h1>
      </div>
      <div class="content-text col-md-9">
        <div class='content-inner'>
          <div class='loader'></div>
          </div>
        </div>
      </div>
    </div>
  </div>

  <div id="learn" class="row-outer row-outer-content">
    <div class="row">
      <div class="content-header col-md-3">
        <h1>Opi</h1>
      </div>
      <div class="content-text col-md-9">
        <div class='content-inner'>
          <div class='loader'></div>
          </div>
        </div>
      </div>
    </div>
  </div>

  <div id="now" class="row-outer row-outer-content">
    <div class="row">
      <div class="content-header col-md-3">
        <h1>Ajankohtaista</h1>
      </div>
    </div>
  </div>

```

```
<div class="content-text col-md-9">
  <div class='content-inner'>
    <div class='loader'></div>
    </div>
  </div>
</div>

</div>
</body>
</html>
```

LIITE 2

```
<!DOCTYPE html>

<html>
<head>
<title>Kaikille.fi</title>
<meta charset="UTF-8">
</head>
<body>
<div class="ohjeet" align="center">
  <h1>Internet Explorer versiosi on liian vanha!</h1>
  <p>Tämä web-sivusto ei tue internet explorer versiota 9 tai siitä vanhempia.</p>
  <p>Hanki uusin versio Internet Explorerista täältä: <a href="http://windows.mic-
    rosoft.com/fi-fi/internet-explorer/download-ie">Lataa Internet Explorer.</a></p>
  <br>
  <p>Muita selaimia löydät näistä linkeistä:</p>
  <p>Hanki uusin versio Internet Explorerista täältä: <a
    href="https://www.google.com/intl/fi-fi/chrome/browser/">Lataa Chrome.</a></p>
  <p>Hanki uusin versio Internet Explorerista täältä: <a
    href="http://www.mozilla.org/fi/firefox/new/">Lataa Firefox.</a></p>
  <br>
  <p>Jos olet osa organisaatiota ota yhteys selaimen päivittämiseen liittyvissä
    asioissa organisaatiosi tietohallintoon/järjestelmänvalvojaan.</p>
</div>
</body>
</html>
```

```

var functions = {};

functions.f = function() {
}

$(document).ready(function(){

    /*Lista kategorioista ja niiden WordPress osoitteista*/
    var elem_ids = {
        "2": "work",
        "6": "school",
        "52": "home",
        "69": "freetime",
        "73": "learn",
        "77": "now",
        "75": "find",
        "71": "ask",
        "1614": "intro"
    };

    /*script_dont_mod.js <--älä muokkaa*/
    UIfunctions();

    /*Funktio sivuston yhden kohdan sisällön muuttamiseen*/
    function formFunction(wanted_id, class_id){

        $('<div class="menu-' + class_id + '"><li><a href="#"></a></li></div>').find("li").find("a").unbind("click").click(function(e){
            e.preventDefault();

            var linkHref = $(this).attr('href');
            var linkSave = linkHref;

            linkHref = linkHref.split("page_id=");
            linkHref = linkHref[1];

            $.ajax({
                type: "GET",
                url: "backend_wp.php",
                data: {
                    get_one_post: linkHref,
                    class: class_id
                },
                cache: false
            })
            .done(function( data ) {
                data=$.parseJSON(data);

                var titl = "<div class='data-block data-title'>" + data.title + "</div>";
                var cont = "<div class='data-block data-content'><div class='data-block-inner'>" + data.content + "</div></div>";
                var comm = "<div class='data-block data-comments'><div class='comment-button'>Kommentoi</div><div class='data-block-inner closed'>" + data.comments + "</div></div>";

                if(wanted_id == linkHref){
                    var append_this = titl + cont;
                }
                else{
                    var append_this = titl + cont + comm;
                }
            });
        });
    }
}

```

```

$afterThis = $('#'+data.id);

$('.appended-'+data.id).remove();
$afterThis.find('.content-inner').append("<div class='appended-
"+class_id+"'>" + append_this + "</div>");

dynamicUI();

var refreshButton = "<button id='PageRefresh'>Lataa Kaikille.fi uudel-
leen.</button>";

$afterThis.find('form').submit( function() {
    $.ajax( {
        type: "POST",
        url: $afterThis.find('form').attr('action'),
        data: $afterThis.find('form').serialize(),
        success: function( response ) {

            /*Code to process the response*/
            $afterThis.find('#respond').html("<br><br><b>Viestin lähetys onnis-
tui.</b><br> Kiitos palauttestasi. Jos viestisi ei vielä näy se odot-
taa tarkistusta.").append("<br>" + refreshButton);

            /*Sivu jälleen ladataan viestin kirjoituksen jälkeen*/
            $('#PageRefresh').click(function(){
                location.reload(true);
            });
        }
    });
    return false;
});
})
.done(function( data ) {
    data=$.parseJSON(data);

    $afterThis = $('#'+data.id);

    var replyRhef = $('#.comment-reply-link').attr('href');
    var testiRhef = $('#.comment-reply-link');

    dynamicUI();

    $('#.comment-reply-link').click(function(e){
        e.preventDefault();
    });
});
});
}

function loadContent(wanted_id,class_id){
    $.ajax({
        type: "GET",
        url: "backend_wp.php",
        data: {
            get_one_post: wanted_id,
            class: class_id },
        cache: false
    })
    .done(function( data ) {
        $el_id = $("#"+class_id);
        $el_class = $(". "+class_id);

        data=$.parseJSON(data);

        $el_id.find(".content-header").find("h1").html("").append(data.thumbnail);

```



```

var titl = "";
var cont = "<div class='data-block data-content'><div class='data-block-inner'>"+data.content+"</div></div>";
var comm = "<div class='data-block data-comments'><div class='comment-button'>Kommentoi</div><div class='data-block-inner closed'>"+data.comments+"</div></div>";
var menu = "<div class='data-block data-menu'><div class='data-block-inner menu-' +class_id+'>"+data.menu+"</div></div>";
var size = "<div class='font'><div class='font-size-' +class_id+'><a id='normal' href='#' class='current'>A</a><a id='large' href='#' class='>A</a></div></div>";

var append_this = titl + cont;

append_this = "<div class='appended-' +class_id+'>"+append_this+"</div>";

if (data.menu.length>1) {
    $el_id.find(".content-header").find("h1").after("<div class='menu-bar'>"+menu+"</div>");
}

$el_id.find(".content-text").find(".content-inner").html("").append("<h1>"+data.title+"</h1>").append(append_this);
$el_id.find(".content-inner").find("h1").append(size);

dynamicUI();          /*Function call for showing and hiding comment section*/
formFunction(wanted_id, class_id);
textSize();           /*Function call for text size changer*/
accordionMenu();      /*Function call for accordion menu*/
});
}

$.each(elem_ids, function( index, value ) {
    loadContent(index,value);
});
});

```

```

function UIFunctions(){

    function runUI(){
        $(".fix a[href^='#'], .navigation a[href^='#']").unbind('click').on('click',
        function(e) {
            e.preventDefault();

            $('html, body').animate({ scrollTop: $(this.hash).offset().top -
            $('.fix').height()}, 300);
        });
    }
    runUI();

    $(window).scroll(function(){
        var bottom = $('.navigation').position().top + $('.navigation').outer-
        Height(true);
        bottom -= $('.fix').height();

        var scrollpos = $(window).scrollTop();

        if(scrollpos >= bottom){
            $('.fix').html("").append($('.navigation').html()).addClass('active');
            runUI();
        }
        else{
            $('.fix').removeClass('active');
            runUI();
        }
    });
}

function dynamicUI(){
    $('.comment-button').unbind('click').click(function(e){
        e.preventDefault();
        $el = $(this).next();

        if(!$el.hasClass("open")){
            $(this).html('Sulje');
            $el.addClass('open');
        }
        else{
            $(this).html('Kommentoi');
            $el.removeClass('open');
        }
    });
}

function accordionMenu(){
    $(document).ready( function($) {
        var sub = false;

        // This shows all the sub menus on page load
        $('.menu ul.children').show();
        // This hides all the sub sub menus on page load
        $('.menu ul ul ul').hide();
        // This makes sure that all the sub menus are open when applicable
        $('.menu li.current-menu-item').children().show();
        // This keeps the nav open to the item that you are navigating to.
        $('.menu li.current-menu-item').parents().show();

        $('.menu ul li a').click(

```

```

function() {
    var checkElement = $(this).next();
    if((checkElement.is('ul')) && (checkElement.is(':visible')) {
        return false;
    }
    if((checkElement.is('ul')) && (!checkElement.is(':visible'))) {
        $(this).parent().siblings("li:has(ul)").find("ul").slideUp('fast');

        checkElement.slideDown('fast');
        return false;
    }
}
);
});
}

```

```

function textSize(){
    $normal = $('#normal');
    $large = $('#large');

    $normal.click(function(e) {
        e.preventDefault();
        $(this).addClass('current');
        $large.removeClass('current');
        $("p").css("font-size", "16px");
        $("li").css("font-size", "16px");
        $(".content-inner h1").css("font-size", "36px");
    });
    $large.click(function(e) {
        e.preventDefault();
        $normal.removeClass('current');
        $(this).addClass('current');
        $("p").css("font-size", "20px");
        $("li").css("font-size", "20px");
        $(".content-inner h1").css("font-size", "45px");
    });
}

```

```

function getInternetExplorerVersion(){
    var rv = -1;
    if (navigator.appName == 'Microsoft Internet Explorer'){
        var ua = navigator.userAgent;
        var re = new RegExp("MSIE ([0-9]{1,})[\\.\0-9]{0,})");
        if (re.exec(ua) != null)
            rv = parseFloat(RegExp.$1);
    }
    else if (navigator.appName == 'Netscape'){
        var ua = navigator.userAgent;
        var re = new RegExp("Trident/.*rv:([0-9]{1,})[\\.\0-9]{0,})");
        if (re.exec(ua) != null)
            rv = parseFloat(RegExp.$1);
    }
    return rv;
}

```

```

<?php
    if( isset($_GET['get_one_post']) ){
        $id = $_GET['get_one_post'];

        $content=file_get_contents('http://testi1.kaikille.fi/backend.php?get_all_con-
tent=' . $id);
        $comments=file_get_contents('http://testi1.kaikille.fi/backend.php?get_com-
ments=' . $id);
        $title=file_get_contents('http://testi1.kaikille.fi/backend.php?get_titles=' .
$id);
        $thumbnail=file_get_contents('http://testi1.kaikille.fi/backend.php?get_thumb-
nails=' . $id);
        $menu=file_get_contents('http://testi1.kaikille.fi/backend.php?get_menu=' . $id);

        $arr=array(
            'title' => $title,
            'thumbnail' => $thumbnail,
            'menu' => $menu,
            'comments' => $comments,
            'content' => $content,
            'id' => $_GET['class'],
        );

        echo json_encode($arr);
    }
?>

```

```

<?php
$path = 'wordpress';

require_once($path."/wp-load.php"); //brings stuff from load.php
require_once($path."/wp-includes/post-template.php"); //Brings functions from
post-template.php file from wp-includes folder to ensure title handling.

require_once($path."/wp-includes/category-template.php");
require_once($path."/wp-includes/post-thumbnail-template.php"); //Brings func-
tions for thumbnail handling.
require_once($path."/wp-includes/comment.php");
require_once($path."/wp-includes/media.php");
require_once($path."/wp-includes/query.php"); //Brings functions for Query
require_once($path."/wp-includes/comment.php");
require_once($path."/wp-includes/comment-template.php");
require_once($path."/wp-includes/functions.php");
require_once($path."/wp-includes/category-template.php");
require_once($path."/wp-includes/nav-menu-template.php");

if( isset($_GET['get_thumbnails']) ){
    //GET's argument in get_all, can be accessed from URL for example
    backend.php?get_all=6.
    $parent = $_GET['get_thumbnails']; //Stores the argument to variable.

    /*Array for the WP_Query*/
    $args = array(
        'p' => $parent, //Variable is stored to array.
        'post_type' => 'page' //Defines the type of the shown object, page,
                                post, any.
    );

    $the_query = new WP_Query( $args ); //Works, brings info from array.

    /*The Loop starts here*/
    if ( $the_query->have_posts() ) :
    while ( $the_query->have_posts() ) : $the_query->the_post();

        // Your loop code
        $feat_image = get_the_post_thumbnail($parent, 'thumbnail');
        echo $feat_image;

```

```

        endwhile;
    else :
        echo wpautop( 'Sorry, no posts were found' ); //Error message.
    endif;
}

if( isset($_GET['get_titles']) ){
    //GET's argument in get_all, can be accessed from URL for example
    backend.php?get_all=6.*/
    $parent = $_GET['get_titles']; //Stores the argument to variable.*/

    /*Array for the WP_Query*/
    $args = array(
        'p' => $parent, //Variable is stored to array.*/
        'post_type' => 'page' //Defines the type of the shown object, page,
                                post, any.*/
    );

    $the_query = new WP_Query( $args ); /*Works, brings info from array.*/

    /*The Loop starts here*/
    if ( $the_query->have_posts() ) :
        while ( $the_query->have_posts() ) : $the_query->the_post();

            /*Your loop code*/
            the_title();

        endwhile;
    else :
        echo wpautop( 'Sorry, no posts were found' ); /*Error message.*/
    endif;
}

if( isset($_GET['get_all_content']) ){
    //GET's argument in get_all, can be accessed from URL for example
    backend.php?get_all=6.
    $parent = $_GET['get_all_content']; //Stores the argument to variable.

    /*Array for the WP_Query*/
    $args = array(
        'p' => $parent, //Variable is stored to array.
        'post_type' => 'page' //Defines the type of the shown object, page,
                                post, any.
    );

```

```

);

$the_query = new WP_Query( $args ); //Works, brings info from array.

/*The Loop starts here*/
if ( $the_query->have_posts() ) :
while ( $the_query->have_posts() ) : $the_query->the_post();

    /*Your loop code*/
    the_content();

endwhile;
else :
    echo wpautop( 'Sorry, no posts were found' ); //Error message.
endif;
}

//Comment handling starts here
if( isset($_GET['get_comments']) ){
    //GET's argument in get_all, can be accessed from URL for example
    backend.php?get_all=6.
    $parent = $_GET['get_comments']; //Stores the argument to variable.

    /*Array for the WP_Query*/
    $args = array(
        'p' => $parent, //Variable is stored to array.
        'post_type' => 'page' //Defines the type of the shown object, page,
                                post, any.
    );

    $parameters = array(
        'post_id' => $parent, //Variable is stored to array.
        //'status' => 'approve',
        //'post_type' => 'page' //Defines the type of the shown object, page,
                                post, any.
    );

    $the_query = new WP_Query( $args ); //Works, brings info from array.

    /*The Loop starts here*/
    if ( $the_query->have_posts() ) :
    while ( $the_query->have_posts() ) : $the_query->the_post();

```

```

// Your loop code
//Retrieve a list of approved comments.
$comment_array = get_approved_comments($parameters);

//Tuo kommentit ja tiedot sekä nappulat.
wp_list_comments( $parameters, $comment_array );

comment_form( $parameters, $parent ); //Tuo kommentti formin esille.

endwhile;
else :
    echo wpautop( 'Sorry, no posts were found' ); //Error message.
endif;
}

//Reply handling starts here
if( isset($_GET['get_reply']) ){
    //GET's argument in get_all, can be accessed from URL for example
    backend.php?get_all=6.
    $parent = $_GET['get_reply']; //Stores the argument to variable.

    /*Array for the WP_Query*/
    $args = array(
        'p' => $parent, //Variable is stored to array.
        'post_type' => 'page' //Defines the type of the shown object, page,
                                post, any.
    );

    $parameters = array(
        'title_reply_to' => $parent,
    );

    $the_query = new WP_Query( $args ); //Works, brings info from array.

    /*The Loop starts here*/
    if ( $the_query->have_posts() ) :
        while ( $the_query->have_posts() ) : $the_query->the_post();
            $comment = $parent;

            // Your loop code
            comment_reply_link($comment);

        endwhile;
    }
}

```



```

else :
    echo wpautop( 'Sorry, no posts were found' ); //Error message.
endif;
}

/*Dropdown menu handling starts here*/
if( isset($_GET['get_menu']) ){
    /*GET's argument in get_all, can be accessed from URL for example
    backend.php?get_all=6.*/
    $parent = $_GET['get_menu'];          /*Stores the argument to variable.*/

    /*This makes an array for every page that is a parent.*/
    $pageIdArray = get_pages(
        array(
            'parent' => 0,
        )
    );

    /*Array for the WP_Query*/
    $args = array(
        'p' => $parent,                /*Variable is stored to array.*/
        'post_type' => 'page',         /*Defines the type of the shown object, page,
                                        post, any.*/
    );

    $the_query = new WP_Query( $args ); /*Works, brings info from array.*/

    /*The Loop starts here*/
    if ( $the_query->have_posts() ) :
    while ( $the_query->have_posts() ) : $the_query->the_post();
        /*Your loop code*/

        $args=array(
            'child_of' => $parent
        );
        $pages = get_pages($args);
        if ($pages) {
            $pageids = array();
            foreach ($pages as $page) {
                $pageids[] = $page->ID;
                $idCount = count($pageids);
            }

            $args=array(

```

```

        'title_li' => 'Tree of Parent Page ' . $parent,
        'include' => $parent . ',' . implode(",", $pageids)
    );

    wp_page_menu($args);

}
else {
    /*echo "null";*/
}

$arr = array(
    'testi' => $idCount
);

endwhile;
else :
    echo wpautop( 'Sorry, no posts were found' ); /*Error message.*/
endif;
}

?>

```

```

@import url(http://fonts.googleapis.com/css?family=Open+Sans:400,700,300);
@import url("menu.css");

body{
    color: white;
}
h1{
    font-family: 'Open Sans', sans-serif;
}
p{
    max-width: 45em !important;
    font-size: 16px;
    text-align: left !important;
}

/*OVERRIDERS start*/
.col-xs-1, .col-xs-2, .col-xs-3, .col-xs-4, .col-xs-5, .col-xs-6, .col-xs-7, .col-
xs-8, .col-xs-9, .col-xs-10, .col-xs-11, .col-xs-12, .col-sm-1, .col-sm-2, .col-sm-
3, .col-sm-4, .col-sm-5, .col-sm-6, .col-sm-7, .col-sm-8, .col-sm-9, .col-sm-10,
.col-sm-11, .col-sm-12, .col-md-1, .col-md-2, .col-md-3, .col-md-4, .col-md-5, .col-
md-6, .col-md-7, .col-md-8, .col-md-9, .col-md-10, .col-md-11, .col-md-12, .col-lg-
1, .col-lg-2, .col-lg-3, .col-lg-4, .col-lg-5, .col-lg-6, .col-lg-7, .col-lg-8, .col-
lg-9, .col-lg-10, .col-lg-11, .col-lg-12 {
    padding: 0;
}
.container{
    max-width: 100%;
    margin: 0;
    padding: 0;
    overflow-x: hidden;
}
a{
    color: white;
}
a:hover{
    color: black !important;
}
.row {
    margin: 0px;
}
/*OVERRIDERS end*/

```

```

/*transitions 0.25*/
.t1,a,body{
  -webkit-transition: 0.25s ease-out;
  -moz-transition: 0.25s ease-out;
  -o-transition: 0.25s ease-out;
  transition: 0.25s ease-out;
}
/*transitions 0.5*/
.t2{
  -webkit-transition: 0.5s ease-out;
  -moz-transition: 0.5s ease-out;
  -o-transition: 0.5s ease-out;
  transition: 0.5s ease-out;
}

/*cursors*/
.home,
.school,
.work,
.freetime,
.ask,
.find,
.learn,
.now,
.comment-button{
  cursor: pointer;
}

.row-outer{
  min-height: 200px;
  width:100%;
}
.row-outer-header {
  background-image: url('../con_img/layout/ylasivun_aalto.png');
}

.navigation{
  overflow: hidden;
}

.row-outer-nav{
  background-color: transparent;
}
.nav-header{

```

```

        text-align: center;
    }
    .nav-header-inner h1 {
        line-height: 200px;
    }
    /*row1*/
    .home .nav-header-inner, #home.row-outer-content, .school .nav-header-inner,
    #school.row-outer-content, .work .nav-header-inner, #work.row-outer-content,
    .freetime .nav-header-inner, #freetime.row-outer-content{
        background-color:#1f78b4;
    }
    .home .nav-header-inner:hover, .school .nav-header-inner:hover, .work .nav-header-
    inner:hover, .freetime .nav-header-inner:hover{
        background-color:orange;
    }
    /*row2*/
    .ask .nav-header-inner, #ask.row-outer-content, .find .nav-header-inner, #find.row-
    outer-content, .learn .nav-header-inner, #learn.row-outer-content, .now .nav-header-
    inner, #now.row-outer-content{
        background-color:#1f78b4;
    }
    .ask .nav-header-inner:hover, .find .nav-header-inner:hover, .learn .nav-header-in-
    ner:hover, .now .nav-header-inner:hover{
        background-color:orange;
    }
    /*Content area coloring*/
    #intro.row-outer-content,#school.row-outer-content,#freetime.row-outer-con-
    tent,#find.row-outer-content,#now.row-outer-content{
        background-color:white;
        color: black;
    }
    #intro.row-outer-content a,#school.row-outer-content a,#freetime.row-outer-content
    a,#find.row-outer-content a,#now.row-outer-content a{
        color: black;
    }
    /*content*/
    .content-header h1 {
        line-height: 200px;
        text-align: center;
    }
    .content-inner{
        margin: auto;
        width: 100%;
    }

```

```

/*find*/
#find .content-inner{
    margin: auto;
    width: 100%;
}
#gmap-here{
    height: 400px;
    width: 100%;
}
/*find*/
#now .content-inner{
    margin: auto;
    width: 100%;
}

/*navigation*/
.fix.active{
    background-color: white;
}
.fix {
    position: fixed;
    width: 100%;
    top: 0px;
    left: 0px;
    height: auto;
    overflow: hidden;
    z-index: 10;
    display: none;
}
.fix.active{
    display: block;
}
.nav-header{
}
.fix .nav-header{
    box-shadow: 0px 0px 10px -3px black inset;
}
.fix .nav-header:hover{
    background-color: orange;
}
.fix .nav-header h1 {
    line-height: 20px;
    font-size: 24px;
    color: black;
}

```

```

}

.fix .row-outer {
    min-height: 0px;
    width: 100%;
}

.fix .nav-header-inner{
    background-color: transparent;
    box-shadow: 0px 0px 0px 0px transparent inset;
}

/*main nav*/
.nav-header-inner {
    margin: auto;
    width: 90%;
    position: relative;
    border-radius: 20px;
    box-shadow: 2px 2px 10px rgb(129, 129, 129);
}

/*header*/
.header-inner {
    width: 200px;
    margin: auto;
    height: 200px;
    background-color: black;
    border-radius: 100px;
    box-shadow: 2px 2px 10px rgb(129, 129, 129);
    top: -10px;
    position: relative;
}

.row-outer-header{

}

.header{
    text-align: center;
}

.header h1 {
    font-size: 30px;
    line-height: 200px;
}

/*loader*/
.loader{

```

```

width: 50px;
height: 50px;
background-color: white;
padding: 5px;
border-radius: 5px;
margin: auto;
position: relative;
top: 20px;
}
.loader img{
width:100%;
}

/*content*/
.content-text h1{
width: 80%;
margin: auto;
margin-top: 20px;
margin-bottom: 20px;
}
.data-content {
background-color: rgba(192, 192, 192, 0.1);
}
.data-content a{
color: orange !important;
}
.data-block.data-comments {
background-color: rgba(0, 0, 0, 0.06);
}
.data-block-inner {
position: relative;
margin: auto;
width: 80%;
padding-top: 20px;
padding-bottom: 20px;
}
.data-block-inner a:hover{
color: black;
}

/*comments close*/
.data-comments .data-block-inner.closed{
height: 0px;
overflow: hidden;

```



```

padding-top: 0px;
padding-bottom: 0px;
}
.data-menu{
    /*vertical-align: middle;*/
}
.open{
    height: auto !important;
    padding: 20px 0px !important;
}
.comment-button {
    width: 220px;
    background: gray;
    color: white;
    text-align: center;
    font-size: 30px;
    line-height: 70px;
}

/*Font-size changer*/
.font {
    float: right;
    position: relative;
}
.font-size a#normal {
    font-size: 25px;
    margin-right: 10px;
}
.font-size a#large {
    font-size: 40px;
    margin-right: 10px;
}
.font-size a.current {
    text-decoration: underline;
}
.font-size-work a#normal, .font-size-school a#normal, .font-size-home a#normal,
.font-size-freetime a#normal, .font-size-learn a#normal, .font-size-now a#normal,
.font-size-find a#normal, .font-size-ask a#normal, .font-size-intro a#normal {
    font-size: 25px;
    margin-right: 10px;
}
.font-size-work a#large, .font-size-school a#large, .font-size-home a#large, .font-
size-freetime a#large, .font-size-learn a#large, .font-size-now a#large, .font-size-
find a#large, .font-size-ask a#large, .font-size-intro a#large {

```

```

    font-size: 40px;
    margin-right: 10px;
}
.font-size-intro a.current, .font-size-home a.current {
    text-decoration: underline;
}
/*testing for infobox*/
div .info {
    min-width: 400px;
}
.luokka {
    background-color: red;
}
/*Mobile screen css*/
@media (max-width: 640px) {
    .fix .nav-header h1{
        font-size: 10px;
        font-weight: bold;
        margin-top: 10px;
        margin-bottom: 0px;
        line-height: 10px;
    }
    .content-header{
        display: flex;
        justify-content: center;
        /*align-items: center;*/
        vertical-align: middle;
    }
    p img{
        max-width: 100%;
        height: auto;
    }
    p iframe{
        width: 100%;
        height: auto;
    }
    div .gmap-here {
        width: 100%;
        height: auto;
    }
}
}

```

```
.menu,
.menu ul,
.menu ul li,
.menu ul li a {
    margin: 0;
    padding: 0;
    border: 0;
    list-style: none;
    line-height: 1;
    display: block;
    position: relative;
    -webkit-box-sizing: border-box;
    -moz-box-sizing: border-box;
    box-sizing: border-box;
    z-index: 9;
}
.menu {
    width: 200px;
    font-family: Helvetica, Arial, sans-serif;
    letter-spacing: 1px;
    color: #ffffff;
    margin-left: auto;
    margin-right: auto;
}
.menu a {
    color: white !important;
}
.menu ul ul {
    display: none;
}
.align-right {
    float: right;
}
.menu > ul > li > a {
    padding: 15px 20px;
    border-left: 1px solid #32373e;
    border-right: 1px solid #32373e;
    border-top: 1px solid #32373e;
    cursor: pointer;
    z-index: 2;
    font-size: 14px;
    text-decoration: none;
```

```

color: #ffffff;
text-shadow: 0 1px 1px rgba(0, 0, 0, 0.35);
background: #606f83;
background: -webkit-linear-gradient(to bottom, #4775D1 0%, #204060 100%);
background: -moz-linear-gradient(to bottom, #4775D1 0%, #204060 100%);
background: -o-linear-gradient(to bottom, #4775D1 0%, #204060 100%);
background: -ms-linear-gradient(to bottom, #4775D1 0%, #204060 100%);
background: linear-gradient(to bottom, #4775D1 0%, #204060 100%);
box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.15);
}

.menu > ul > li > a:hover,
.menu > ul > li.active > a,
.menu > ul > li.open > a {
    color: #eeeeee;
    background: #FFC966;
    background: -webkit-linear-gradient(to bottom, #FFC966, #FF8C00);
    background: -moz-linear-gradient(to bottom, #FFC966, #FF8C00);
    background: -o-linear-gradient(to bottom, #FFC966, #FF8C00);
    background: -ms-linear-gradient(to bottom, #FFC966, #FF8C00);
    background: linear-gradient(to bottom, #FFC966, #FF8C00);
}

.menu > ul > li.open > a {
    box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.15), 0 1px 1px rgba(0, 0, 0, 0.15);
    border-bottom: 1px solid #CAD3D3;
}

.menu > ul > li:last-child > a,
.menu > ul > li.last > a {
    border-bottom: 1px solid #8D9494;
}

.holder {
    width: 0;
    height: 0;
    position: absolute;
    top: 0;
    right: 0;
}

.holder::after,
.holder::before {
    display: block;
    position: absolute;
    content: '';
    width: 6px;
    height: 6px;
    right: 20px;

```

```

    z-index: 10;
    -webkit-transform: rotate(-135deg);
    -moz-transform: rotate(-135deg);
    -ms-transform: rotate(-135deg);
    -o-transform: rotate(-135deg);
    transform: rotate(-135deg);
}
.holder::after {
    top: 17px;
    border-top: 2px solid #ffffff;
    border-left: 2px solid #ffffff;
}
.menu > ul > li > a:hover > span::after,
.menu > ul > li.active > a > span::after,
.menu > ul > li.open > a > span::after {
    border-color: #eeeeee;
}
.holder::before {
    top: 18px;
    border-top: 2px solid;
    border-left: 2px solid;
    border-top-color: inherit;
    border-left-color: inherit;
}
.menu ul ul li a {
    cursor: pointer;
    border-bottom: 1px solid #32373e;
    border-left: 1px solid #32373e;
    border-right: 1px solid #32373e;
    padding: 10px 20px;
    z-index: 1;
    text-decoration: none;
    font-size: 13px;
    color: #eeeeee;
    background: -webkit-linear-gradient(to bottom, #69717c 0%, #3d4552 100%);
    background: -moz-linear-gradient(to bottom, #69717c 0%, #3d4552 100%);
    background: -o-linear-gradient(to bottom, #69717c 0%, #3d4552 100%);
    background: -ms-linear-gradient(to bottom, #69717c 0%, #3d4552 100%);
    background: linear-gradient(to bottom, #69717c 0%, #3d4552 100%);
    box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.1);
}
.menu ul ul li:hover > a,
.menu ul ul li.open > a,
.menu ul ul li.active > a {

```

```

background: #FFC966;
background: -webkit-linear-gradient(to bottom, #FFC966, #FF8C00);
background: -moz-linear-gradient(to bottom, #FFC966, #FF8C00);
background: -o-linear-gradient(to bottom, #FFC966, #FF8C00);
background: -ms-linear-gradient(to bottom, #FFC966, #FF8C00);
background: linear-gradient(to bottom, #FFC966, #FF8C00);
color: #ffffff;
}
.menu ul ul li:first-child > a {
    box-shadow: none;
}
.menu ul ul ul li:first-child > a {
    box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.1);
}
.menu ul ul ul li a {
    padding-left: 30px;
    background: -webkit-linear-gradient(to bottom, #3C3F46 0%, #1E2227 100%);
    background: -moz-linear-gradient(to bottom, #3C3F46 0%, #1E2227 100%);
    background: -o-linear-gradient(to bottom, #3C3F46 0%, #1E2227 100%);
    background: -ms-linear-gradient(to bottom, #3C3F46 0%, #1E2227 100%);
    background: linear-gradient(to bottom, #3C3F46 0%, #1E2227 100%);
}
.menu > ul > li > ul > li:last-child > a,
.menu > ul > li > ul > li.last > a {
    border-bottom: 0;
}
.menu > ul > li > ul > li.open:last-child > a,
.menu > ul > li > ul > li.last.open > a {
    border-bottom: 1px solid #32373e;
}
.menu > ul > li > ul > li.open:last-child > ul > li:last-child > a {
    border-bottom: 0;
}
.menu ul ul li.has-sub > a::after {
    display: block;
    position: absolute;
    content: '';
    width: 5px;
    height: 5px;
    right: 20px;
    z-index: 10;
    top: 11.5px;
    border-top: 2px solid #eeeeee;
    border-left: 2px solid #eeeeee;

```

```
-webkit-transform: rotate(-135deg);
-moz-transform: rotate(-135deg);
-ms-transform: rotate(-135deg);
-o-transform: rotate(-135deg);
transform: rotate(-135deg);
}
.menu ul ul li.active > a::after,
.menu ul ul li.open > a::after,
.menu ul ul li > a:hover::after {
    border-color: #ffffff;
}
.menu ul ul ul ul li a {
    background: #d4edfe;
    padding-left: 45px;
}
.menu ul ul ul ul ul li a {
    background: white;
}
```